

This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1. (Currently Amended) A computerized system for software development comprising:
 - a source code editor operable to edit a source code module;
 - a graphical design surface operable to display a graphical object representing actual code of the source code module and provide integrated testing of the source code module, wherein the integrated testing includes load testing;
 - a change manager operative to manage versioning of the source code module;
 - ~~and~~
 - an application datastore operative to store a previous version of the source code module; and
 - a package manager operative to provide an interface adapted for highlighting a set of software modules to be grouped together as a package and further adapted for receiving properties to be associated with a package,wherein upon a change in the source code module, the change in the source code is immediately communicated to the graphical design surface and the graphical design surface is updated to reflect the change in the source code module, wherein the design surface displays the graphical object, the graphical object represents a database object, the design surface is operative to bind a particular database system to the database object, the database object further includes a database column, the source code module includes a variable, and the design surface is operative to bind the database column to the variable.
2. (Original) The computerized system of claim 1, wherein upon a change in the graphical design surface, the change in the graphical design surface is immediately communicated to the source code editor and the source code editor is updated to reflect the change in the graphical design surface.
3. (Cancelled)

4. (Previously Presented) The computerized system of claim 1, wherein a difference between the source code module and the previous version of the source code module is highlighted by source code editor.

5. (Original) The computerized system of claim 4, wherein the difference is highlighted using a squiggly line under the difference.

6. (Original) The computerized system of claim 4, wherein the difference is highlighted using a tooltip bar to indicate a date and an author of the difference.

7. (Previously Presented) The computerized system of claim 1, wherein a difference between the source code module and the previous version of the source code module is highlighted by the design surface.

8. (Original) The computerized system of claim 1, further comprising at least one compiler operative to compile the source code module into an object code format.

9. (Original) The computerized system of claim 1, wherein the design surface is operative to bind the source code module to the at least one compiler.

10-11. (Canceled)

12. (Original) The computerized system of claim 1, wherein the binding is established through a drag-and-drop interface.

13. (Cancelled)

14. (Original) The computerized system of claim 13, wherein the package manager is operative to receive a list of system identifiers, each of the system identifiers

identifying a particular computer system, and wherein the package manager is further operative to provide an interface to determine the particular system to deploy the package to.

15. (Currently Amended) A computerized method for developing a software project, the method comprising:

creating a graphical object on a design surface, the graphical object representing actual code of a software module;

binding the graphical object to an application type;

generating source code particular to the application type;

maintaining versioning data of the software module; and

storing a previous version of the software module;

receiving identification of a set of software modules to be grouped together as a package;

receiving properties to be associated with a package.

wherein the design surface displays the graphical object, the graphical object represents a database object, the design surface is operative to bind a particular database system to the database object, the database object further includes a database column, the source code module includes a variable, and the design surface is operative to bind the database column to the variable, wherein the design surface further provides support for integrated testing of the software module, further wherein the integrated testing includes load testing.

16. (Original) The computerized method of claim 15, wherein the application type is a source code compiler.

17. (Original) The computerized method of claim 15, wherein the application type is a database application.

18. (Original) The computerized method of claim 15, wherein the application type is a source code interpreter.

19. (Original) The computerized method of claim 15, further comprising:
modifying the source code; and
refreshing the design surface to update the graphical object to reflect the
modification to the source code.
20. (Original) The computerized method of claim 15, further comprising:
modifying the graphical object on the design surface; and
refreshing the source code to reflect the modification to the graphical object.
21. (Original) The computerized method of claim 15, further comprising reading
a template having pre-configured software modules from a datastore.
22. (Currently Amended) A computer-readable medium having computer
executable instructions for performing a method for developing a software project, the
method comprising:
creating a graphical object on a design surface, the graphical object
representing actual code of a software module;
binding the graphical object to an application type in response to a user input
selecting an application type;
generating source code particular to the application type;
maintaining versioning data of the software module; ~~and~~
storing a previous version of the software module;
in response to a user input, creating a package populated with different
software components;
receiving a user input identifying properties to be associated with the package;
in response to a user input, creating a deployment group comprising a
collection of packages for deployment;
presenting a matrix of machines and defined deployment groups;
receiving a user input mapping a deployment group to a set of machines; and
storing the mapping of a deployment group to a set of machines,

wherein the design surface displays the graphical object, the graphical object represents a database object, the design surface is operative to bind a particular database system to the database object, the database object further includes a database column, the source code module includes a variable, and the design surface is operative to bind the database column to the variable, wherein the design surface is further operative to provide integrated testing of the software module, further wherein the integrated testing includes load testing.

23. (Original) The computer-readable medium of claim 22, wherein the application type is a source code compiler.

24. (Original) The computer-readable medium of claim 22, wherein the application type is a database application.

25. (Original) The computer-readable medium of claim 22, wherein the application type is a source code interpreter.

26. (Original) The computer-readable medium of claim 22, wherein the method further comprises:

- modifying the source code; and
- refreshing the design surface to update the graphical object to reflect the modification to the source code.

27. (Original) The computer-readable medium of claim 22, wherein the method further comprises:

- modifying the graphical object on the design surface; and
- refreshing the source code to reflect the modification to the graphical object.

28. (Original) The computer-readable medium of claim 22, wherein the method further comprises reading a template having pre-configured software modules from a datastore.